

CS:3330 Algorithms

Midterm Exam (100 points)

Closed books and notes (except one sheet of notes)

None of the digital devices is allowed.

3/26/2019

- (30 points) We insert the following numbers in the given order into an empty binary search tree. (a) If the tree is an AVL tree, for each rotation in an insertion, please display the tree before and after rotation. If no rotation happens during an insertion, simply draw the tree after the insertion. Please repeat the above task when the tree is (b) a Red-Black tree, or (c) a splay tree:
2, 9, 7, 4, 3, 5.
- (30 points) We like to sort the array $A = [2, 9, 7, 4]$, and the only operation to change the array A is by $\text{swap}(i, j)$, which swaps the elements of A at positions i and j .
(a) If heapsort is used, please illustrate how the heap is constructed and changed by the swap operation (show the content of the array A after each swap operation).
(b) If quicksort is used (and the first element of the subarray is the pivot), please show the content of the array A after each swap operation.
- (40 points) Given two sets A and B , $\text{union}(A, B)$ will create a new set C consisting of elements from A and B with cost $O(|A| + |B|)$, where $|X|$ is the size of X , i.e., the number of elements in set X .
(a) Please design an efficient algorithm (as fast as you can) which unions n disjoint sets of the same size m , into a single set by calling $\text{union}(A, B)$. Please analyze the complexity of your algorithm in terms of n and m .
(b) If the sizes of these n disjoint sets are $1, 2, \dots, n$, respectively, please repeat the question in (a).

Answer:

- (a) We may use the idea of merge sort to union all sets into one. Let $\mathbf{S} = \{ S_1, S_2, \dots, S_n \}$, each S_i is a set of m elements. The pseudo code of the algorithm is given as follows:

```
unionAll(S)
  let  $\mathbf{S} = \{ S_1, S_2, \dots, S_n \}$ 
  if  $n = 1$  return  $S_1$ 
   $d = (n+1)/2$ 
   $A = \text{unionAll}(\{S_1, S_2, \dots, S_d\})$ 
   $B = \text{unionAll}(\{S_{d+1}, \dots, S_n\})$ 
  return  $\text{union}(A, B)$ 
```

Let $T(n, m)$ be the complexity of $\text{unionAll}(\mathbf{S})$. Then $T(n, m) = 2T(n/2, m) + cmn$, where cmn is the cost of $\text{union}(A, B)$ and c is a constant. The solution is $O(mn \log(n))$.

(b) Without loss of generality, we assume n is even (if not, we add an empty set). At first, we pair the set of size 1 to the set of size n , the set of size 2 to the set of size $n-1$, and so on. We then call $\text{union}(A, B)$ to union each pair of sets into one set. The whole time takes $O(n/2(n+1))$, or $O(n^2)$. The result \mathbf{R} is a set of $n/2$ sets of size $n+1$. Then we call $\text{unionAll}(\mathbf{R})$. From (a), we know the complexity is $O(mn/2 \log(n/2))$, where $m = n+1$. So the total complexity is $O((n+1)n/2 \log(n/2) + n^2)$, or $O(n^2 \log(n))$. The pseudo code is:

```
unionAll2(S)
  let  $\mathbf{S} = \{ S_1, S_2, \dots, S_n \}$ 
   $\mathbf{R} = \{ \}$ 
  for  $k$  from 1 to  $n/2$ 
     $\mathbf{R} = \mathbf{R} \cup \{ \text{union}(S_k, S_{n-k+1}) \}$ 
  return  $\text{unionAll}(\mathbf{R})$ 
```